

Projet LO02

Shape Up !

Il vous est proposé dans ce projet de concevoir avec UML et développer en Java une version électronique du jeu de cartes « Shape Up ! ».

1. Règles du jeu

Les règles du jeu vous sont données en annexe. Elles servent de document de référence pour l'expression des besoins du logiciel développé.

Au-delà, on considère aussi les besoins suivants :

- L'application devra permettre de jouer à deux ou trois joueurs qui seront des joueurs physiques auxquels s'ajoutent des joueurs virtuels. Dans le cas d'un joueur virtuel, le jeu devra lui attribuer une stratégie de jeu simple mais cohérente qui lui permettra de déterminer quelles actions mener à chaque instant.
- L'ensemble de l'application sera intégré dans une interface graphique. On pourra utiliser des images telles que celles données en annexe.
- Avant la conception et le développement de l'interface graphique, une interface rudimentaire en ligne de commande permettra de tester le moteur du jeu. Cette interface devra être conservée lors de l'évaluation fonctionnelle et la remise des fichiers.
- On développera le moteur du jeu de manière à exploiter la généricité offerte par la programmation orientée objet. Cela se traduira par :
 - Des variantes dans les règles du jeu. En début de partie, le choix d'une variante sera proposé au joueur et c'est la variante choisie qui imposera les règles d'une partie. Il est demandé de concevoir et implémenter deux variantes en plus des règles de base.
 - Le choix de la forme du tapis de jeu où les cartes sont posées. La règle impose un rectangle de 5 cartes par 3. Vous proposerez d'autres formes de tapis de jeu (triangle, disque, etc.) qui pourront être choisis en début de partie.
- L'architecture retenue devra veiller au respect des règles de la conception orientée objet. L'architecture devra ainsi être (1) modulaire en identifiant des composants indépendants liés entre eux par des relations et (2) extensible en permettant le changement de règles ou l'ajout de cartes avec de nouvelles fonctions.

2. Phases et jalons

Le projet sera découpé en trois phases qui conduiront à des soutenances effectuées durant une séance de TP dédiée. Les trois phases du projet sont :

1. la modélisation UML initiale ;
2. le développement du moteur du jeu, utilisable en lignes de commandes ;
3. le développement de l'interface graphique et la remise du code source documenté.

Pour chaque phase et jalon associé, les consignes à suivre vous sont données ci-après.

2.1. Phase 1 : modélisation UML

La modélisation UML proposée sera exposée sous la forme d'une présentation qui devra suivre la structure suivante :

1. Introduction : présentation du projet tel que vous l'avez compris (pas de copier/coller de l'énoncé) et annonce du plan du document
2. Diagramme de cas d'utilisation : identifier l'ensemble des cas d'utilisation du système, leurs relations et les expliquer.
3. Diagramme de classes : décrire et expliquer chaque élément de conception du diagramme. Par ailleurs, vous indiquerez l'étude que vous aurez faite sur (1) le patron de conception *Strategy*, utilisé en autres, pour les joueurs virtuels, et (2) le patron *Visitor*, utilisé pour le comptage des scores. Vous préciserez la manière dont vous l'intégrez dans votre modélisation. Vous noterez qu'il est inutile de modéliser les interfaces de commande du jeu (en lignes de commandes ou graphique) ; on ne représentera que le cœur de l'application.
4. Diagramme de séquence : Proposer un diagramme de séquence pour le déroulement d'un tour de jeu.
5. Conclusion : Identifier les aspects sûrs de la modélisation et ceux dont vous pensez que le développement pourrait induire une modification.

2.2. Phase 2 : cœur de l'application et interface en lignes de commandes

Cette phase, validée par une seconde soutenance orale, consistera à faire une première démonstration du développement. A cette étape, le moteur du jeu devra être fonctionnel et utilisable en lignes de commandes. Mais aucune mise en forme particulière du code produit n'est demandée (mise en packages, documentation, mise au propre du code, ...). De même, aucun support de présentation ne devra être préparé pour cette seconde soutenance.

La généricité de forme du tapis de jeu devra être intégrée par l'implémentation d'une interface de collection choisie parmi les interfaces existantes de la plateforme Java (par ex. Set, List, Map, Queue). Les différentes implémentations faites de cette interface permettront de faire varier la forme du tapis.

2.3. Phase 3 : projet complet et documentation

Cette dernière phase consistera à mettre en œuvre la version finale du projet ainsi que toute la documentation qui l'accompagne. D'un point de vue du travail de développement, il consistera à intégrer une interface graphique au cœur de l'application développée précédemment. Attention, le jeu devra rester jouable à la fois en ligne de commandes et par le biais de l'interface graphique, au sein d'une même partie. Pour ce faire, vous devrez mettre en œuvre le patron de conception MVC, vu en cours en considérant que l'application possède deux vues concurrentes.

Une soutenance orale permettra de valider les aspects fonctionnels de l'application et une remise des fichiers du projet permettra d'évaluer le code, son organisation, sa structure et sa documentation.

Evaluation fonctionnelle

Elle sera effectuée par le biais d'une soutenance qui aura lieu durant votre dernière séance de TP. Elle s'effectue par binôme et dure 15 minutes durant lesquelles vous présenterez l'état fonctionnel de votre application par le biais d'une démonstration. Une fois la démonstration passée, tout développement supplémentaire (correction de bugs, ajout de fonctionnalités, ...) est inutile car l'état fonctionnel de votre projet sera évalué à ce moment. Suite à cela, quelques questions relatives au code que vous avez produit vous seront posées.

Remarque : Pour cette évaluation fonctionnelle, il ne vous est pas demandé de mettre au propre le code présenté (respect des conventions d'écriture, commentaire Javadoc, suppression des blocs de codes inutiles mis en commentaires, ...). Par contre, cela devra être fait pour la remise des fichiers.

Remise des fichiers du projet

Les fichiers du projet à remettre seront déposés sur Moodle, dans un espace dédié à cet effet, et placés dans une archive ZIP dont le nom du fichier doit suivre la convention de nommage suivante :

<Nom1>_<Nom2>_ProjetLO02_A20.ZIP, où <Nom1> et <Nom2> sont données par ordre alphabétique. Par exemple : Doyen_Langeron_ProjetLO02_A20.PDF

L'archive devra comporter les dossiers suivants :

- **src** : contiendra l'ensemble des sources du projet ;
- **dist** : contiendra un fichier nommé projet.jar qui contiendra une archive JAR exécutable du projet. C'est ce fichier qui sera exécuté pour tester si besoin votre projet après la soutenance ;
- **classes** : contiendra l'ensemble des classes compilées du projet ;
- **doc** : contiendra la javadoc du projet, le diagramme de classes final et l'état fonctionnel de l'application..

Enfin, les consignes à observer pour la mise en forme du code développé dans le cadre du projet sont les suivantes :

- L'ensemble du code doit être documenté par le biais de commentaires javadoc. Il ne faudra pas se contenter des balises standard mais décrire précisément la fonction de chaque élément de code (classe, méthode, ...) en respectant les conventions de documentation pour la production logicielle telles que vues durant le semestre.
- Le code devra être propre et suivre les conventions d'écriture spécifiées par Oracle (nommage, indentation, blocs, casse, ...). Aucun code obsolète, placé dans un commentaire, ne devra figurer dans les fichiers sources.
- Le diagramme de classes final. On détaillera ici et justifiera les changements entre la version initiale du diagramme de classes et la version finale, effectivement implémentée dans le code. Pour ce faire, on présentera les deux diagrammes.
- L'état actuel de l'application : Cette partie donnera de manière précise l'état de l'application en regard du présent cahier des charges. On y indiquera clairement ce qui a été implanté et ce qu'il ne l'est pas, ce qui fonctionne et ce qui reste buggé.

3. Consignes générales

3.1. Calendrier

Semaine	Evénement
2 (21/09)	Distribution des projets et constitution des binômes
6 et 8 (19/10 et 2/11)	Soutenance Jalon 1
11 et 12 (23/11 et 30/11)	Soutenance Jalon 2
17 et 18 (4/01 et 11/01)	Soutenance
19 (vendredi 22/01 à 23:55)	Remise des fichiers

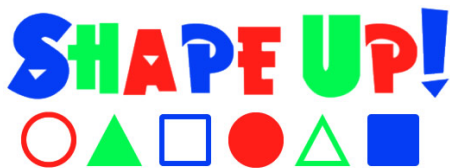
3.2. Respect des droits d'auteurs

Le présent projet consiste en la modélisation et le développement logiciel d'un jeu dont les conditions d'utilisation sont spécifiées ici : <http://www.goodlittlegames.co.uk/about-faq.html>. L'Université de Technologie de Troyes n'utilise cette œuvre que dans seul un cadre d'enseignement

et à des fins exclusivement pédagogiques. En effectuant ce travail au sein d'une unité d'enseignement, l'étudiant s'engage à respecter les droits d'auteurs et d'exploitation liés à tout support de stockage et de diffusion numérique du code produit dans le cadre de son projet et à respecter le cadre exclusivement pédagogique associé à cette production. Toute autre forme d'exploitation et de diffusion par un étudiant relève de sa seule responsabilité individuelle.

3.3. *Autres consignes*

- Pour l'ensemble des livrables du projet, aucun retard ne sera accepté et aucune raison ne pourra le justifier.
- Vos intervenants de TP sont vos interlocuteurs exclusifs pour toute question relative au projet.



Mo Holkar

2 or 3 players • 10 minutes

Circles, triangles or squares? Solid or hollow? Red, green or blue? Your challenge is to play cards to fill the grid and line up the shapes that will score you the most points — while trying to stop your opponent doing exactly the same thing!

SETUP

Shuffle all the cards and then deal as follows:

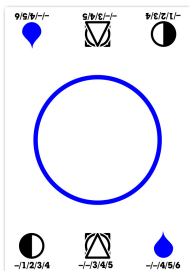
- **Hidden card:** Deal one card face down beside the play area, without looking at it. This card is out of play and will not be used during the game.
- **Victory cards:** Deal each player one victory card face down. Keep your victory card hidden. It tells you which lines of symbols will score you points at the end of the game.

Place the remaining cards face down beside the playing area to form the deck.

Choose a start player in any way you wish.

Changes for 3 players are at the end of the rules.

VICTORY CARDS



This card shows the symbol: **blue hollow circle**.

If this card was your victory card, you would score points at the end of the game for each line of blue symbols, for each line of hollow symbols, and for each line of circles.

AIM OF THE GAME

The aim of the game is to play and move cards to create a card layout so that your victory card will score the most points at the end of the game.

HOW TO PLAY

The start player goes first. Then take alternating turns. At the start of your turn, **draw the top card** from the deck and look at it.

Then take the following actions, in either order:

- **Play your card (mandatory):** You must play your card face up into the playing area, adding it to the card layout.
- **Move a card (optional):** You may move an existing card to a new position within the card layout. If you choose to move a card, you may do so before or after you play your card.

Placement rules

Every time you play or move a card you must follow both of these placement rules:

1. **Adjacency:** You may only play or move a card into a position that is orthogonally adjacent to at least one other card already in play. Cards are not adjacent if they only touch at their corners.

On the first turn there are no other cards are in play, so ignore the adjacency rule. Simply lay a card in the middle of the play area to begin the layout.

2. **Card layout size:** After you play or move a card, all of the cards in play must lie within an imaginary grid of cards that is 5 cards long and 3 cards wide (or 3 cards long and 5 cards wide). The borders of the grid are not fixed at the start of the game, so in your first few turns you can play or move cards into any position.

Moving cards

When you move an existing card, you may break the layout of cards into two or more groups of cards, or leave single cards in positions in which they are no longer adjacent to any other cards. The card you move, however, must follow the adjacency rule.

END OF THE GAME

The game ends when the last card in the deck has been drawn and played, and the grid of cards is complete.

Scoring

Now reveal your victory card and count up your score. The player with the highest total score wins.

You score points for every horizontal or vertical line of symbols within the card layout that matches the scoring opportunities on your victory card. Diagonal lines do not score, but a single card may score twice as part of two overlapping lines of symbols.

		1	2	3	4	5	6
Circle Triangle Square		-	1	2	3	4	
Filled Hollow		-	-	3	4	5	
Red Green Blue		-	-	4	5	6	

This table shows the number of points you receive for each line of symbols that match one of the attributes of the symbol on your victory card. Longer lines score more points. This scoring information is repeated on the top and bottom of every card.

Scoring example

In this example, Player A scores 5 different lines for 13pts, and Player B scores 3 lines for 10pts.

Player A's victory card

3 hollows: 3pts

3 greens: 4pts

2 circles: 1pt

3 circles: 2pts

3 hollows: 2pts

Score 13pts

Player B's victory card

4 reds: 6pts

3 triangles: 2pts

3 filled: 3pts

Score 10pts

PLAYING A FULL GAME

A full game of *Shape Up!* is played over four rounds. Alternate the start player and add up the scores from each round to determine the overall winner.

OTHER WAYS TO PLAY

Shape Up! for 3 players

Three players can play the game in exactly the same way, taking turns clockwise. The only difference is that, after dealing the hidden card and one victory card to each player, there will be 14 cards in the deck. This means that the final card layout will have one empty position.

Advanced Shape Up!

In the advanced game, do not deal out victory cards at the start of the game. Instead, you will choose for yourself which of the cards that pass through your hand is your victory card.

At the start of the game, deal each player a hand of three cards.

On your turn, you must play one card from your hand. At the end of your turn, draw one from the deck to replace (skip this step once the deck is used up).

At the end of the game you will have one card remaining in your hand: this card is your victory card.

Many thanks to Tracy Bose, Dom Camus, Alex Churchill, Matt Dunstan, Roger Gammans, Katarina Harris, Helen Pritchard, Behrooz Shahriari, Selena Timmins Chapman, Cat Tobin, and all the other playtesters!



Rules v1.0 • © 2013 Mo Holkar
www.goodlittlegames.co.uk

